accenture

Learning and Knowledge Management

Mastering Ansible

Testing Ansible Playbooks

Copyright © 2020 Accenture. All Rights Reserved.

Version	Revision Date	Description	Author(s)	Reviewed by	Approved by
V1.0	14-09-2020	Initial Version	Ranjith Kumar Thirumalai Ramesh	Rathnajyothi Perumalsamy Vishnu Kallimakula	CPCL Sreenivasa Rao

Copyright © 2020 Accenture. All Rights Reserved.

Index

Copyright © 2020 Accenture. All Rights Reserved.

Exercise: Testing Ansible Playbook

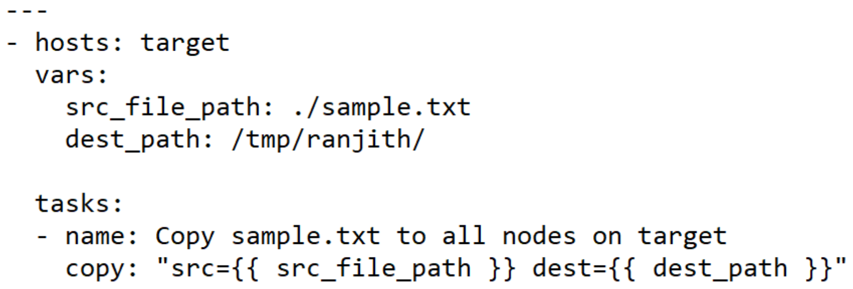
Prerequisite

Scenario: *In this activity, we will cover how to test the ansible playbooks with various strategies.*

Sub Activity 1: Working with --syntax-check, --check and --diff options of playbook execution

<table>
<tr>
<td><b>Step 1</b></td>
<td>

Create a **playbook.yml** file with the below content –



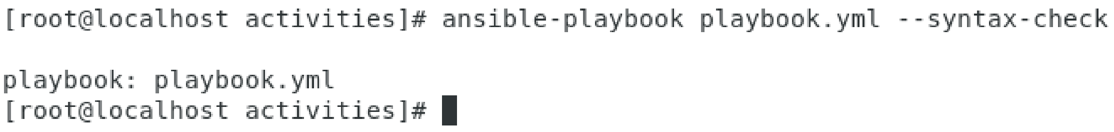
```
---
- hosts: target
  vars:
    src_file_path: ./sample.txt
    dest_path: /tmp/ranjith/

  tasks:
  - name: Copy sample.txt to all nodes on target
    copy: "src={{ src_file_path }} dest={{ dest_path }}"
```

</td>
</tr>
<tr>
<td><b>Step 2</b></td>
<td>

Execute the below command to check if there are any syntax-errors.

ansible-playbook playbook.yml --syntax-check

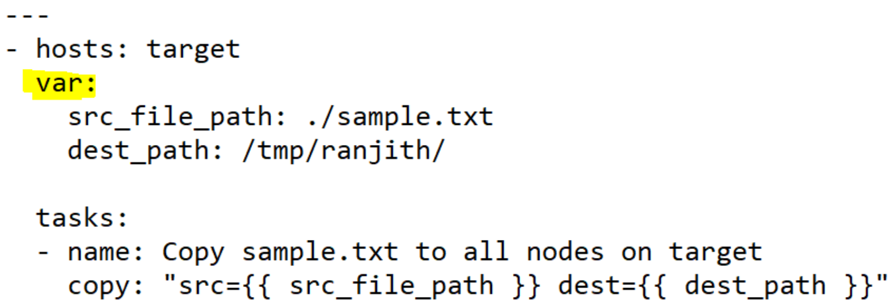
If it is not having any syntax-errors, the result would be as shown below –



```
[root@localhost activities]# ansible-playbook playbook.yml --syntax-check

playbook: playbook.yml
[root@localhost activities]# 
```

</td>
</tr>
<tr>
<td><b>Step 3</b></td>
<td>

Create a syntax-error by changing a keyword “vars” as “var” as shown below and execute the same command.

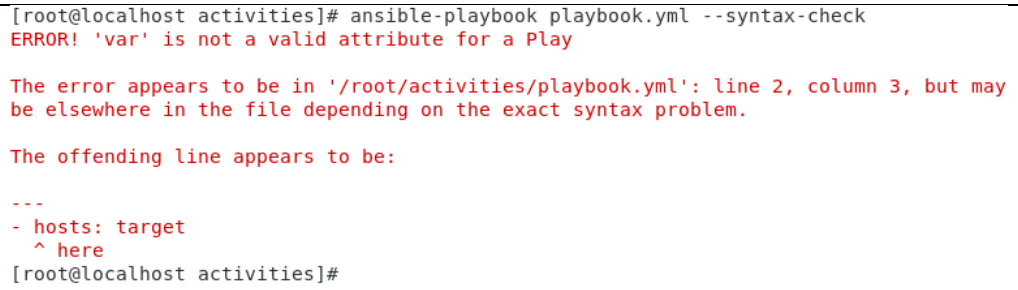


```
---
- hosts: target
  var:
    src_file_path: ./sample.txt
    dest_path: /tmp/ranjith/

  tasks:
  - name: Copy sample.txt to all nodes on target
    copy: "src={{ src_file_path }} dest={{ dest_path }}"
```

Since you have an error now, the error message would be as shown below –

</td>
</tr>
</table>

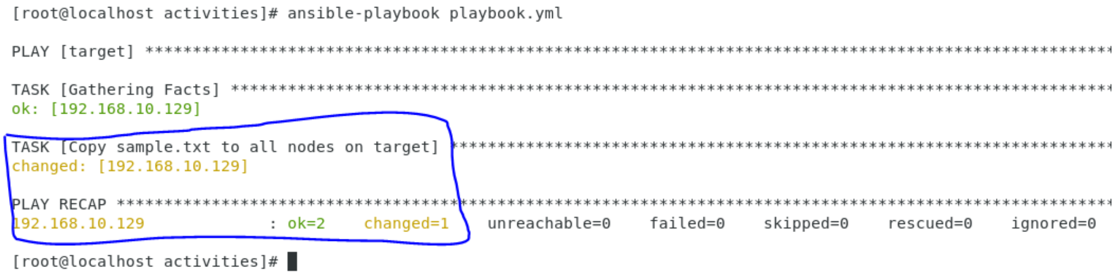
Copyright © 2020 Accenture. All Rights Reserved.

<table>
<tr><td></td><td>

<pre>
[root@localhost activities]# ansible-playbook playbook.yml --syntax-check
ERROR! 'var' is not a valid attribute for a Play

The error appears to be in '/root/activities/playbook.yml': line 2, column 3, but may
be elsewhere in the file depending on the exact syntax problem.

The offending line appears to be:

- hosts: target
  ^ here
[root@localhost activities]#
</pre>
</td></tr>
<tr><td><b>Step 4</b></td><td>Correct the created mistake and execute the syntax check command once again to confirm if the syntax error is corrected.</td></tr>
<tr><td><b>Step 5</b></td><td>Execute the below command to execute the playbook.

<b>ansible-playbook playbook.yml</b>
This will execute and show the output as shown below –

<pre>
[root@localhost activities]# ansible-playbook playbook.yml

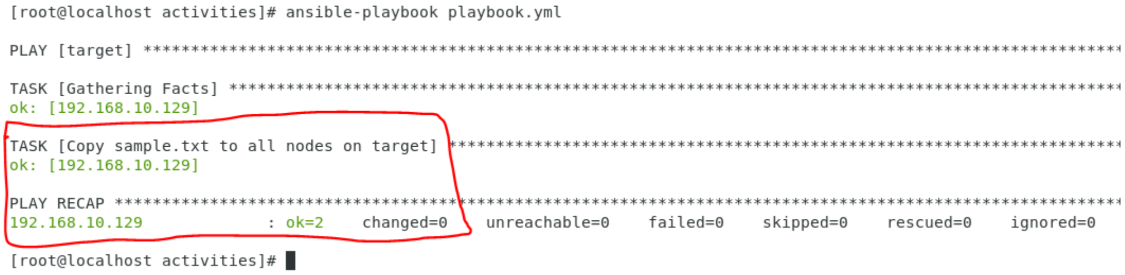
PLAY [target] ***

TASK [Gathering Facts] **
ok: [192.168.10.129]

TASK [Copy sample.txt to all nodes on target] ***
changed: [192.168.10.129]

PLAY RECAP **
192.168.10.129   : ok=2   changed=1   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0

[root@localhost activities]# █
</pre>
<b>Note:</b>
Create <b>sample.txt</b> file in current directory

Create a directory with name <b>ranjith</b> on target node inside <b>tmp</b> directory</td></tr>
<tr><td><b>Step 6</b></td><td>If you execute the same command once again, it will show the execution is OK, but nothing is changed, as there is no change in the content of the sample.txt file or there is no change in the <b>src</b> and <b>dest</b> arguments of <b>copy</b> module. This is because, most of the modules are idempotent in nature. That is, if nothing is there to change in the destination, it remains as it is.

<pre>
[root@localhost activities]# ansible-playbook playbook.yml

PLAY [target] ***

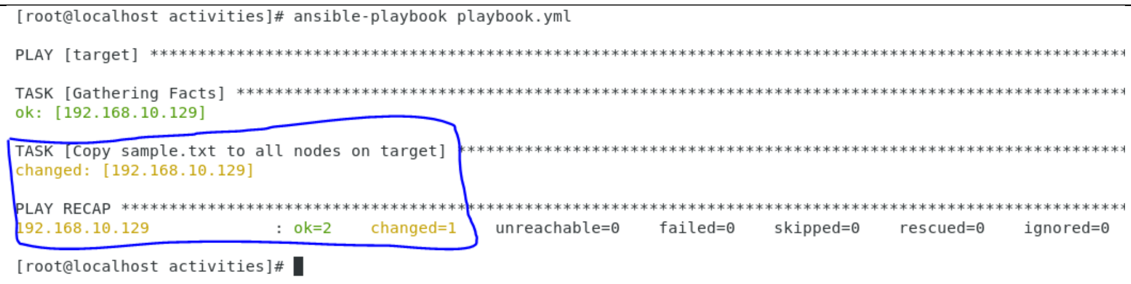
TASK [Gathering Facts] **
ok: [192.168.10.129]

TASK [Copy sample.txt to all nodes on target] ***
ok: [192.168.10.129]

PLAY RECAP **
192.168.10.129   : ok=2   changed=0   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0

[root@localhost activities]# █
</pre>
</td></tr>
<tr><td><b>Step 7</b></td><td>Let us modify the file, sample.txt. You may add some content or delete some content or modify the existing text and execute the ansible-playbook command once again. This results in the above results once again.</td></tr>
</table>

Copyright © 2020 Accenture. All Rights Reserved.



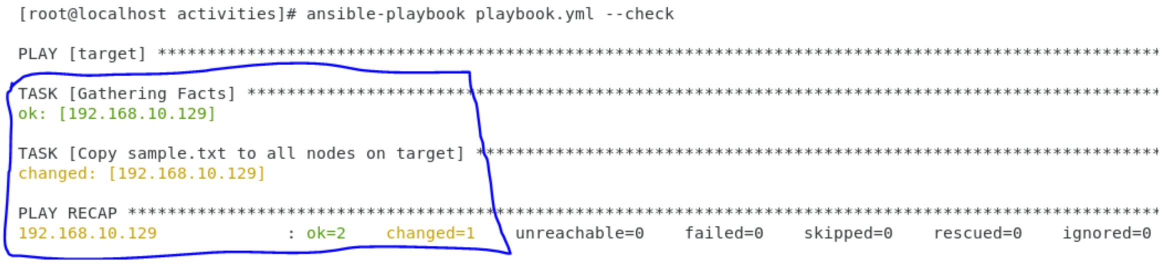
```
[root@localhost activities]# ansible-playbook playbook.yml

PLAY [target] ******************************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.10.129]

TASK [Copy sample.txt to all nodes on target] **********************************************************
changed: [192.168.10.129]

PLAY RECAP *********************************************************************************************
192.168.10.129             : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[root@localhost activities]# 
```

Step 8

In this step let us see, if the ansible-playbook command modifies anything in the destination nodes.
To do as discussed above, modify the sample.txt file once again and see the output. If the output as shown in step 6, it signifies there is no change.

If the output is as shown in step 7, it signifies there is a change in the destination node.

To achieve that, let us modify sample.txt file and execute the below command –
ansible-playbook playbook.yml --check



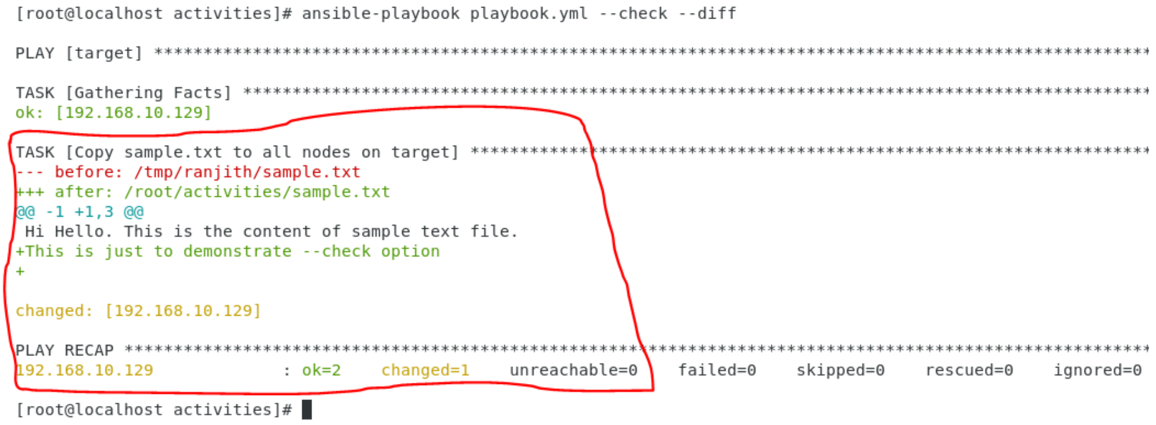
```
[root@localhost activities]# ansible-playbook playbook.yml --check

PLAY [target] ******************************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.10.129]

TASK [Copy sample.txt to all nodes on target] **********************************************************
changed: [192.168.10.129]

PLAY RECAP *********************************************************************************************
192.168.10.129             : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

How many ever times you execute the above command, it results in the same output.

Step 9

If you use --diff option with the above command, it will also show what difference it makes in sample.txt file in the destination node if executed.

ansible-playbook playbook.yml --check --diff



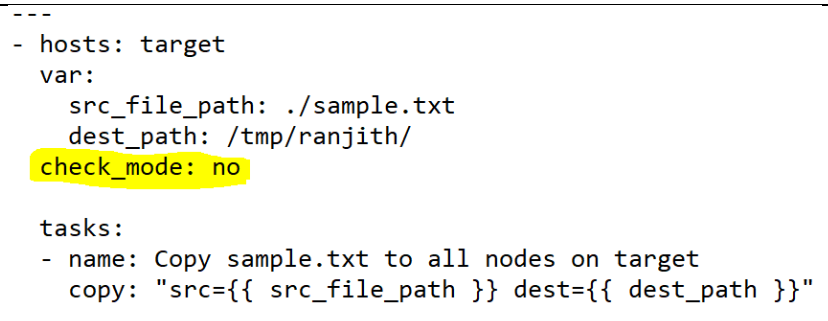
```
[root@localhost activities]# ansible-playbook playbook.yml --check --diff

PLAY [target] ******************************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.10.129]

TASK [Copy sample.txt to all nodes on target] **********************************************************
--- before: /tmp/ranjith/sample.txt
+++ after: /root/activities/sample.txt
@@ -1 +1,3 @@
 Hi Hello. This is the content of sample text file.
+This is just to demonstrate --check option
+

changed: [192.168.10.129]

PLAY RECAP *********************************************************************************************
192.168.10.129             : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[root@localhost activities]# 
```

Step 10

If you add **check_mode** keyword in the **playbook.yml** file as shown in the below image, this will nullify the impact of **--check** option.

Copyright © 2020 Accenture. All Rights Reserved.

<table>
<tr>
<td></td>
<td>



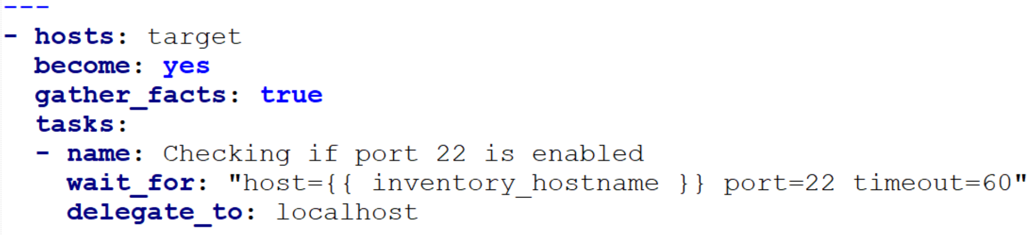
```
---
- hosts: target
  var:
    src_file_path: ./sample.txt
    dest_path: /tmp/ranjith/
  check_mode: no

  tasks:
  - name: Copy sample.txt to all nodes on target
    copy: "src={{ src_file_path }} dest={{ dest_path }}"
```

</td>
</tr>
</table>

Sub Activity 2: Using delegate_to attribute in playbooks

<table>
<tr>
<td><b>Step 1</b></td>
<td>

Create a **playbook.yml** file with the below content –



```
---
- hosts: target
  become: yes
  gather_facts: true
  tasks:
  - name: Checking if port 22 is enabled
    wait_for: "host={{ inventory_hostname }} port=22 timeout=60"
    delegate_to: localhost
```

</td>
</tr>
<tr>
<td><b>Step 2</b></td>
<td>

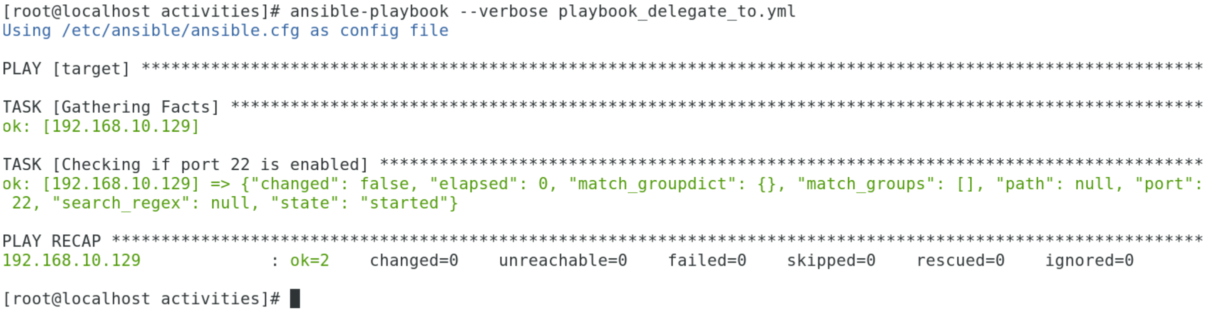
Execute the below command to run the playbook.

ansible-playbook --verbose playbook.yml

</td>
</tr>
</table>

Copyright © 2020 Accenture. All Rights Reserved.

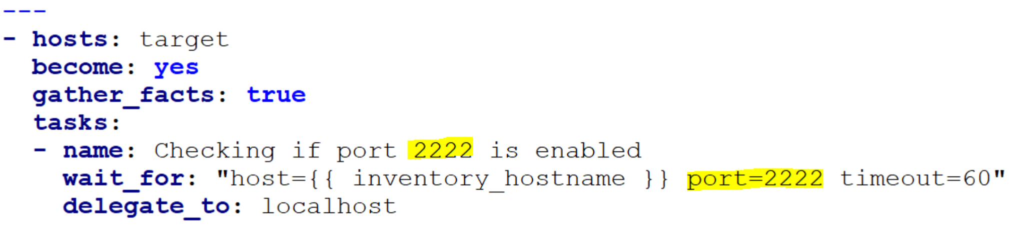
You will get the output as shown below –



```
[root@localhost activities]# ansible-playbook --verbose playbook_delegate_to.yml
Using /etc/ansible/ansible.cfg as config file

PLAY [target] *****************************************************************************************************

TASK [Gathering Facts] ********************************************************************************************
ok: [192.168.10.129]

TASK [Checking if port 22 is enabled] *****************************************************************************
ok: [192.168.10.129] => {"changed": false, "elapsed": 0, "match_groupdict": {}, "match_groups": [], "path": null, "port":
 22, "search_regex": null, "state": "started"}

PLAY RECAP ********************************************************************************************************
192.168.10.129             : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[root@localhost activities]# 
```

Step 3

Modify the port to 2222 instead of 22 (which is not enabled) as shown in the below image –

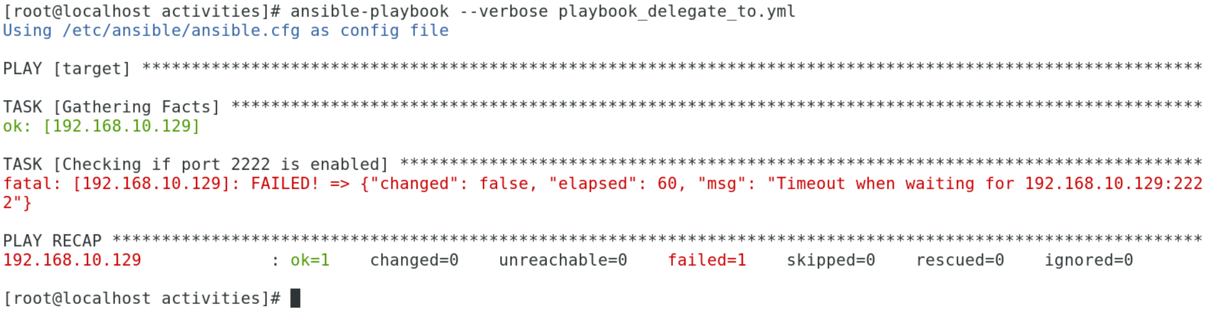


```
---
- hosts: target
  become: yes
  gather_facts: true
  tasks:
  - name: Checking if port 2222 is enabled
    wait_for: "host={{ inventory_hostname }} port=2222 timeout=60"
    delegate_to: localhost
```

After 60 seconds (as mentioned in the **playbook.yml**) Execute the ansible-playbook command.

ansible-playbook --verbose playbook.yml

You will get the output as shown below –

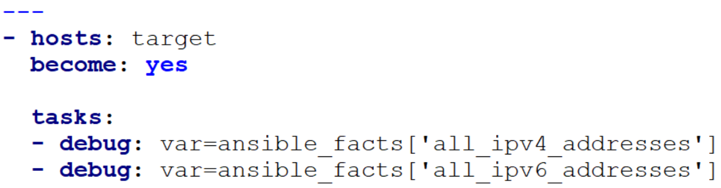


```
[root@localhost activities]# ansible-playbook --verbose playbook_delegate_to.yml
Using /etc/ansible/ansible.cfg as config file

PLAY [target] *****************************************************************************************************

TASK [Gathering Facts] ********************************************************************************************
ok: [192.168.10.129]

TASK [Checking if port 2222 is enabled] ***************************************************************************
fatal: [192.168.10.129]: FAILED! => {"changed": false, "elapsed": 60, "msg": "Timeout when waiting for 192.168.10.129:222
2"}

PLAY RECAP ********************************************************************************************************
192.168.10.129             : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0

[root@localhost activities]# 
```

Sub Activity 3: Using meta module in playbooks

Step 1

Create a **playbook.yml** file with the below content –



```
---
- hosts: target
  become: yes

  tasks:
  - debug: var=ansible_facts['all_ipv4_addresses']
  - debug: var=ansible_facts['all_ipv6_addresses']
```

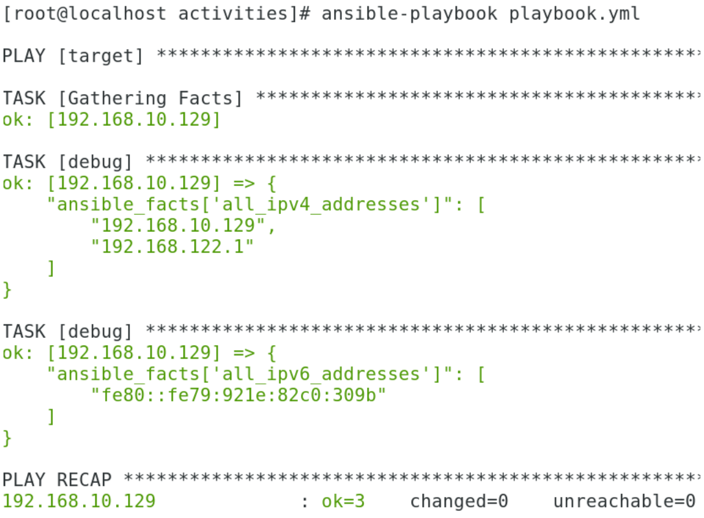
Step 2

Execute the below command to run the playbook.

ansible-playbook playbook.yml

Copyright © 2020 Accenture. All Rights Reserved.

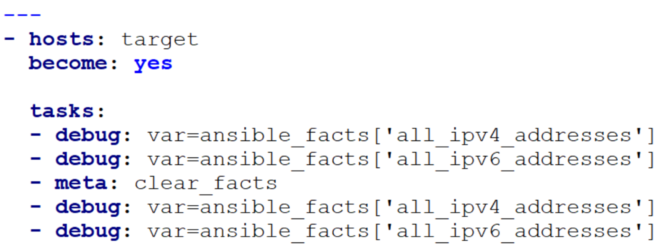
You will get the output as shown below –



```
[root@localhost activities]# ansible-playbook playbook.yml

PLAY [target] ***************************************************

TASK [Gathering Facts] ******************************************
ok: [192.168.10.129]

TASK [debug] ****************************************************
ok: [192.168.10.129] => {
    "ansible_facts['all_ipv4_addresses']": [
        "192.168.10.129",
        "192.168.122.1"
    ]
}

TASK [debug] ****************************************************
ok: [192.168.10.129] => {
    "ansible_facts['all_ipv6_addresses']": [
        "fe80::fe79:921e:82c0:309b"
    ]
}

PLAY RECAP ******************************************************
192.168.10.129             : ok=3    changed=0    unreachable=0
```

Step 3

Modify the **playbook.yml** file with the below content –



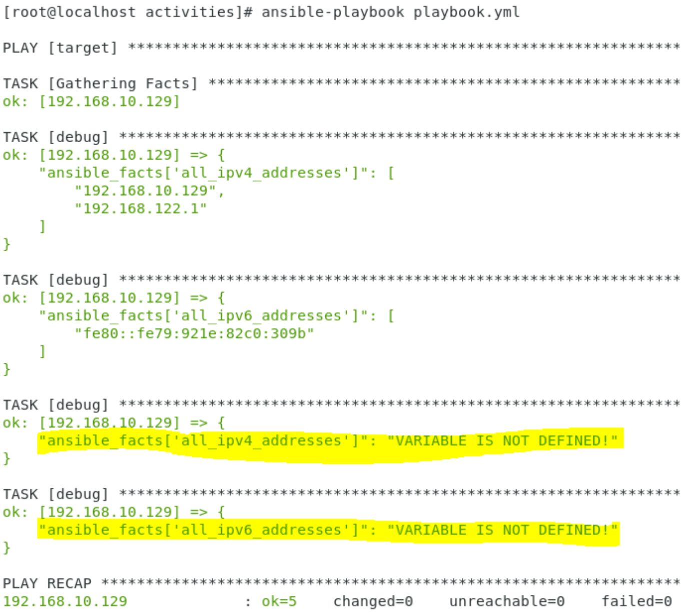
```
---
- hosts: target
  become: yes

  tasks:
  - debug: var=ansible_facts['all_ipv4_addresses']
  - debug: var=ansible_facts['all_ipv6_addresses']
  - meta: clear_facts
  - debug: var=ansible_facts['all_ipv4_addresses']
  - debug: var=ansible_facts['all_ipv6_addresses']
```

Step 4

Execute the below command to run the playbook.

ansible-playbook playbook.yml

You will get the output as shown below –



```
[root@localhost activities]# ansible-playbook playbook.yml

PLAY [target] ***************************************************************

TASK [Gathering Facts] ******************************************************
ok: [192.168.10.129]

TASK [debug] ****************************************************************
ok: [192.168.10.129] => {
    "ansible_facts['all_ipv4_addresses']": [
        "192.168.10.129",
        "192.168.122.1"
    ]
}

TASK [debug] ****************************************************************
ok: [192.168.10.129] => {
    "ansible_facts['all_ipv6_addresses']": [
        "fe80::fe79:921e:82c0:309b"
    ]
}

TASK [debug] ****************************************************************
ok: [192.168.10.129] => {
    "ansible_facts['all_ipv4_addresses']": "VARIABLE IS NOT DEFINED!"
}

TASK [debug] ****************************************************************
ok: [192.168.10.129] => {
    "ansible_facts['all_ipv6_addresses']": "VARIABLE IS NOT DEFINED!"
}

PLAY RECAP ******************************************************************
192.168.10.129             : ok=5    changed=0    unreachable=0    failed=0
```

You can notice in the output, the second set of debug modules, the output says the **ansible_facts** variable is not defined. So, from this it is concluded that the meta module is able to clear the facts that are gathered initially

Copyright © 2020 Accenture. All Rights Reserved.

Copyright © 2020 Accenture. All Rights Reserved.